A MODEL FOR DIALOGUE MANAGEMENT FOR HUMAN COMPUTER INTERACTION

Arne Jönsson

Department of Computer and Information Science
Linköping University, S-581 83, LINKÖPING, SWEDEN
arnjo@ida.liu.se

ABSTRACT

This paper presents a computational model for dialogue management for user-friendly cooperative natural language interaction. The model relies on the assumption that the dialogue can be managed by a dialogue grammar based on speech act information only. Focus structure can be handled through focal parameters that model information in parameters pertaining to a set of salient domain concepts and related predicates ascribed to this set. The focus parameters are modeled in a dialogue tree. Efficiency is achieved by a simple heuristic principle stating that copying relevant information from one segment to the next, updated with information from the application and user initiative, is sufficient for most context dependent utterances. Which action to carry out for task related initiatives is determined from information on how these focal parameters are specified. The dialogue model was originally developed for written interaction, but is also applicable to spoken interaction, as presented in the paper.

1. INTRODUCTION

User-friendly cooperative natural language interaction needs a model of the ongoing dialogue to utilize information on focus and dialogue structure. The module responsible for this is the dialogue manager. A dialogue manager should be designed to be computationally efficient without restricting the user's ability to express herself or noticeably slow down the interaction with the background system. A natural language interface must also clearly show the user which actions it is able to perform, which initiatives it can respond to, which it cannot respond to, and why this is the case [4, 7]. Furthermore, the dialogue manager should be designed to facilitate customization to various applications.

2. THE DIALOGUE MANAGER

The Dialogue Manager presented in this paper is a joint effort of the members of the Natural Language Processing Laboratory at Linköping University, Sweden [3]. It was initially designed for written interaction from an analysis of a corpus of 21 dialogues, collected in Wizard of Oz-experiments using five different background systems.

The Dialogue Manager was developed for written interaction to Simple Service Systems [9]. Such systems require in essence only that the user identifies certain entities, parameters of the service, to the system providing the service. Once they are identified the service can be provided. The Dialogue Manager can be viewed as a controller of resources for interpretation, background system access and generation. It receives input from the interpretation modules, inspects the result and accesses the background system with information conveyed in the user input. Eventually an answer is returned from the database access module and the Dialogue Manager then calls the generation modules to generate an answer to the user. If clarification is needed from any of the resources, it is dealt with by the Dialogue Manager.

The Dialogue Manager uses information from dialogue objects that model the dialogue segments and moves and information associated with them. The dialogue objects represent the constituents of the dialogue and the Dialogue Manager records instances of dialogue objects in a dialogue tree as the interaction proceeds. The dialogue object's parameters reflect the information needed by the Dialogue Manager and the various processes accessing information stored in the dialogue tree. The dialogue object descriptions are domain dependent and can be modified for each new application.

2.1. Dialogue structure

The model for dialogue management assumes that a dialogue is divided into three main classes on the basis of structural complexity. One class corresponding to the size of a dialogue (D), another class to the size of a discourse segment and a third class to the size of a single speech act, or dialogue move. Thus, a dialogue is structured in terms of discourse segments, and a discourse segment in terms of moves and embedded segments. Utterances are not analyzed as dialogue objects, but as linguistic objects which function as vehicles of one or more moves. An initiative-response (IR) structure is assumed where an initiative opens a segment by introducing a new goal and the response closes the segment [8].

There are various other proposals as to the number of categories needed. They differ mainly on the modeling of complex units that consist of sequences of discourse segments, but do not comprise the whole dialogue. For instance, LOKI [15] and SUNDIAL [6] use an intermediate level characterized by having a common topic, i.e. an object whose properties are discussed over a sequence of exchanges. However, a sequence of segments may be connected in a number of different ways; e.g. by dealing with one object for which different properties are at issue. But it may also be the other way around, so that the same property is topical, while different objects are talked about [3].

To specify the functional role of a move, we use the parameters Type and Topic. Type corresponds to the illocutionary type of the move. In simple service systems two sub-goals can be identified [9, p. 266]: 1) "specify a parameter to the system" and 2) "obtain the specification of a parameter". Initiatives are categorized accordingly as being of two different types: 1) update, where users provide information to the system and 2) question, where users obtain information from the system. Responses are categorized as answer, for database answers from the system or answers to clarification requests. Other Type categories are Greeting, Farewell and Discourse Continuation [8], the latter being used for utterances from the system whose purpose is to keep the conversation going.

Topic describes which knowledge source to consult. In information retrieval applications three different topics are used: the database for solving a task, acquiring system-related information about the application, or, finally, the ongoing dialogue. If the background system allows update, e.g. ordering of a specified item, a fourth knowledge source is needed to account for this. A similar knowledge source could contain background information such as current day etc.

2.2. Focus structure

In information retrieval dialogues, the most common user initiative is a request for information from the database. Users specify a database object, or a set of objects, and ask for concept information, e.g. the value of a property of that object or set of objects. Two[1] focal content parameters, termed Objects and Properties, account for the information structure of a move (query), where Objects denote a set of primary referents, and Properties a complex predicate ascribed to this set [1]. The Properties parameter models the domain concept in a sub-parameter termed Aspect which can be specified in another sub-parameter termed Value. These are focal parameters in the sense that they can be in focus over a sequence of IR-units. The values to these parameters depend on the background system, and the natural language interface needs to be customized to account for the demands from each application [12].

The focus structure also needs principles for how the values of the focal parameters Objects and Properties are specified from information in the user initiative and the answer provided from the database. A move can fully specify both Objects and Properties. However, many utterances provide only partial specification of the focal parameters; context information is needed to fully specify them. Two principles account for focus maintenance. A general heuristic principle is that everything not changed in an utterance is copied from one IR-node in the dialogue tree to the newly created IR-node [13, 14]. Another principle is that the value for Objects will be updated with the value from the module accessing the database, if provided. The details of the copying principles need to be customized for each application to meet the demands of the background system and the focal content parameters [12].

2.3. Dynamic behaviour

During the course of interaction, the dialogue tree is built up from instances of dialogue objects with each dialogue object responsible for its own correctness [11]. Initially, the root-node, the D-node, creates an instance of an IR-node and inserts it into the tree, creating links between the IR-node and the D-node. The IR-node creates an instance of a user move which interprets the first move. Upon receiving an instantiated structure from the interpretation modules, the Dialogue Manager determines the Type and Topic of the utterance based on information in the current active node, the IR-unit which initiated the move, and the information given in the current move.

In an information retrieval system the user initiates most segments. The system only takes initiative when needing a clarification from the user. This depends on how the information in the user initiative and the answer from the database system specify the values to the focal parameters. For instance, if the user initiative specifies a correct object but an erroneous property, the dialogue manager records the object specification but asks for a clarification on the property. The action to be carried out for task-related questions can be determined from the specification of the values to the focal parameters Objects and Properties. This, in turn, depends on the user initiative together with the information copied from the previous IR-unit and context information from the dialogue tree and the answer from the database system [13].

There are also user initiatives which do not depend on the values of Objects and Properties, such as system-related questions, i.e. users' requests for information about the system. These are recognized on the grounds of linguistic information provided by the syntactic/semantic analyzer [2].

2.4. Verifying the dialogue model

The Dialogue Manager has been customized[2] for three applications (other than those utilized in the design), one of which is implemented [4], using a set of 30 dialogues, and

[1]For some applications a third parameter, SecondaryObjects, is utilized to constrain the database search [13].

[2]For details on the results from customizing the dialogue and focus structures, see [12].

also verified for these applications [12]. One of the applications, CARS, allows users to retrieve information from a consumers guide on properties of used cars. In another application, TRAVEL, the application domain was charter trips to the Greek archipelago. The TRAVEL application not only utilized information retrieval but also, in one scenario, allowed users to order a specified charter trip.

Customizing the focal parameters Objects and Properties requires an analysis of the application but is often a straightforward task. For a database, the primary referents are the objects for which information is available. A context free grammar with less than 20 rules can accurately model the dialogue structure used in the corpus. The principle of copying information from one dialogue object to the other provides the correct context for most referring expressions. For CARS only 5% required a search in the dialogue tree. The corresponding numbers for TRAVEL were 6% for information retrieval and 2% if ordering is utilized.

The action scheme relating the dialogue structure and focus structure account for most user-initiatives. In the CARS application 85% of the user initiatives are task-related questions. In the TRAVEL application without ordering, the number of task-related user initiatives account for 93% of the user utterances and finally when ordering is allowed 90% of the user utterances are task-related. The other user initiatives are system-related questions, farewells, greetings, etc which are interpreted from linguistic information.

3. APPLYING THE DIALOGUE MODEL TO SPOKEN INTERACTION

The dialogue model has been applied to dialogues from simulations from two different applications utilizing speech interaction. In one application, the SUNDIAL application, users can request flight information via telephone[3]. From this corpus 40 dialogues from four different subjects have been studied. The other application, the Waxholm application, provides information on boat traffic in the Stockholm archipelago [5]. In this application speech input is integrated with text and graphics. From this corpus 10 dialogues were studied.

It is difficult to relate the results of the analysis of the speech applications, Waxholm and SUNDIAL, to the written applications CARS and TRAVEL. One reason is that in Waxholm the Wizard often decides not to understand a user's initiative. This results in a response stating that the system does not understand. Such segments do not pose any serious problem to the Dialogue Manager, the focus structure is not updated and a simple rule accounts for the action.

The design of the system and the properties of the application also influences the analysis, and the user's behaviour [4]. This can be illustrated from how the system responds to utterances which are not specified in enough detail to access the background system. In the CARS and TRAVEL applications the response provides information explaining the system's abilities, without stating any explicit request for new information, e.g.:

> User: *price*
> System: *The price depends on which hotel you choose, when you travel and how long you are away*

i.e., the segment is closed and a new IR-unit initiated. In Waxholm the system instead initiates a clarification request sub-dialogue asking the user to specify the necessary parameters, (cf. frame dialog [10]), e.g.:

> User: *I want to go to Vaxholm tonight*
> System: *Where do you want to leave from?*
> User: *From Stockholm*
> System: *What day of the week do you want to go?*
> User: *On Thursday*

Frame dialog sequences are used in the TRAVEL application where ordering is allowed to collect necessary ordering information, but not otherwise. A difference is that in Waxholm the focal parameters are updated whereas in TRAVEL an order form is utilized.

The SUNDIAL application sometimes provides two moves in one utterance, one providing the requested information and in the next the system asks the user if a certain piece of information should be presented. Such sequences violate the two-part IR-sequence. The system initiates a request to present some information, the user can respond affirmatively to this upon which the system provides the information. Thus, a three-part interaction structure is needed [6]. It is possible to modify the dialogue model to account for such utterances. However, at the moment it has not been done.

There is one further type of phenomenon which is typical for spoken interaction and which is not found in the written interactions – when a user interrupts the system. For instance, the SUNDIAL system always prompts the user with *please wait* to acknowledge the initiative and then the answer follows when the system has accessed the database. Here the user can interrupt, for instance uttering *pardon could you repeat that please* or *thank you*. Such types of interruptions do not add new information and might be ignored; however, it can also be the case that the user modifies her initiative, e.g. *pardon flight two two seven*. This is a type of dynamic behaviour which does not violate the model but the Dialogue Manager's dynamic behaviour needs to be modified to allow the focal parameters to be updated and, a new access to the background system be carried out with the modified request, before an answer is provided.

[3] These dialogues are from a corpus of Wizard of Oz-dialogues collected and transcribed at the Social and Computer Sciences, University of Surrey, UK as part of the ESPRIT Sundial project (P2218).

Taking into account these differences, which mainly depend on the application and the system design, no important deviations arise from the customization of the written applications, CARS and TRAVEL. The SUNDIAL corpus mainly consists of short fully specified task-related user initiatives followed by an answer from the system. Only 12 user initiatives, of 161, require local context, the rest are fully specified. None requires global context. The dialogues in the Waxholm corpus are longer and have more clarification request sequences, but again the principle for managing local focus is sufficient. Of 212 user initiatives, 26 need local context and only 2 need global focus which is even fewer than for the written applications.

However, one phenomenon found in the SUNDIAL corpus is less straightforward. Users sometimes refer to the propositional content of the previous utterance utilizing utterances such as *does that mean ...* where "that" refers to the interpretation of the previous utterance. Such utterances require a complex analysis of the utterance in order to respond appropriately. At the moment we have no simple solution to this. The response will be some type of "Dont understand". On the other hand, this may be the correct answer in order to avoid providing the user with an unrealistic model of the system's capabilities.

4. SUMMARY

This paper presented an efficient dialogue manager for natural language interfaces to Simple Service Systems. It was developed for written interaction, but with minor modifications to account for spoken interaction the dialogue model can also handle phenomena utilized in spoken interaction. The differences that occur are mainly due to the design of the system and the properties of the applications, and show the importance of using empirical material to customize a natural language interface.

5. ACKNOWLEDGMENTS

This work results from a research project supported by The Swedish Council of Research in the Humanities and Social Sciences (HSFR) and The Swedish National Board for Industrial and Technical Development (NUTEK) in the joint Research Program for Language Technology. I am especially indebted to Lars Ahrenberg, Nils Dahlbäck, Magnus Merkel and Åke Thurée.

6. REFERENCES

1. Lars Ahrenberg. *Interrogative Structures of Swedish. Aspects of the Relation between grammar and speech acts.* PhD thesis, Uppsala University, 1987.
2. Lars Ahrenberg. Functional constraints in knowledge-based natural language understanding. In *Proceedings of the 12th International Conference on Computational Linguistics, Budapest*, pages 13–18, 1988.
3. Lars Ahrenberg, Arne Jönsson, and Nils Dahlbäck. Discourse representation and discourse management for natural language interfaces. In *Proceedings of the Second Nordic Conference on Text Comprehension in Man and Machine, Täby, Sweden*, 1990.
4. Lars Ahrenberg, Arne Jönsson, and Åke Thurée. Customizing interaction for natural language interfaces. In *Workshop on Pragmatics in Dialogue, The XIV:th Scandinavian Conference of Linguistics and the VIII:th Conference of Nordic and General Linguistics, Göteborg, Sweden*, 1993.
5. J. Bertenstam, M. Blomberg, R. Carlson, J. Gustafson, S. Hunnicutt, J. Högberg, R. Lindell, L. Neovius, A. de Serpa-Leitao, L. Nord, and N. Ström. Spoken dialogue data collection in the waxholm project. STL-QPSR 1/1995, pages 50–73. 1995.
6. Eric Bilange. A task independent oral dialogue model. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin*, 1991.
7. Nils Dahlbäck and Arne Jönsson. An empirically based computationally tractable dialogue model. In *Proceedings of the Fourteenth Annual Meeting of The Cognitive Science Society, Bloomington, Indiana*, 1992.
8. Nils Dahlbäck. *Representations of Discourse, Cognitive and Computational Aspects.* PhD thesis, Linköping University, 1991.
9. Philip J. Hayes and D. Raj Reddy. Steps toward graceful interaction in spoken and written man-machine communication. *International Journal of Man-Machine Studies*, 19:231–284, 1983.
10. Wolfgang Hoeppner, Katharina Morik, and Heinz Marburger. Talking it over: The natural language dialog system ham-ans. In Leonard Bolc and Matthias Jarke, editors, *Cooperative Interfaces to Information Systems.* Springer-Verlag, Berlin Heidelberg, 1986.
11. Arne Jönsson. A dialogue manager using initiative-response units and distributed control. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin*, 1991.
12. Arne Jönsson. A method for development of dialogue managers for natural language interfaces. In *Proceedings of the Eleventh National Conference of Artificial Intelligence, Washington DC*, pages 190–195, 1993.
13. Arne Jönsson. Dialogue actions for natural language interfaces. In *Proceedings of IJCAI-95, Montréal, Canada*, 1995.
14. Stephanie Seneff. A relaxation method for understanding spontaneous speech utterances. In *Paper presented at the Fifth DARPA Workshop on Speech and Natural Language*, 1992.
15. Tom Wachtel. Pragmatic sensitivity in nl interfaces and the structure of conversation. In *Proceedings of the 11th International Conference of Computational Linguistics, University of Bonn*, pages 35–42, 1986.